Extending Explicit Guidance Methods to Higher Dimensions, Additional Conditions, and Higher Order Integration

Evan Kawamura[1] and Dilmurat Azimov[2]

[1] NASA Ames Research Center, Intelligent Systems Division,
Moffett Field, Mountain View, CA, USA,
evan.t.kawamura@nasa.gov,
[2] University of Hawaii: Manoa, Department of Mechanical Engineering,
2540 Dole Street, Honolulu, HI, USA,
azimov@hawaii.edu

Abstract. Guidance functions play critical roles in autonomy to steer vehicles and aircraft to the intended target or destination. Explicit guidance (E Guidance) solves the two-point boundary value problem with initial and final conditions for position and velocity. The original formulation of E Guidance involves translational acceleration commands with a direct relationship to time, and it is possible to modify E Guidance for rotational acceleration. Other extensions for E Guidance include higher dimensions, additional conditions, and higher-order integration of the linearly independent E Guidance functions. The most promising extension involves higher-order integration of the E Guidance functions, but it may be physically impractical by initially moving away from the target. This paper provides a brief overview of some methods that extend E Guidance to higher dimensions, utilize additional conditions, or perform higher-order integration, and if they satisfy the two-point boundary value problem.

Keywords: explicit guidance, polynomials, unmanned aerial vehicles, boundary conditions, higher-order integration, additional conditions

1 Introduction

Explicit Guidance (E Guidance) started before the famous Apollo missions and became the fundamentals for Apollo lunar guidance. The primary goal for E Guidance involves determining a thrust acceleration vector to solve the rendezvous two-point boundary problem for rocket-propelled spacecraft at a specified terminal time based on the spacecraft's current position and velocity. Even though the original E Guidance derivations were for spacecraft, they can be applied to all types of vehicles [1]. This paper will discuss methods for extending E Guidance: intermediate positions and velocities, desired acceleration and jerk, and higher-order integration of the E Guidance functions [2].

Here is an overview of the E Guidance extensions. The intermediate positions and velocities approach cuts T_{go}, in half and uses desired position and velocity as the third and fourth sets of conditions, which extends $\mathbf{E}$ from $\mathbb{R}^{2\times 2}$ to $\mathbb{R}^{4\times 4}$. Including desired acceleration and jerk resembles Cherry's formulation with final attitude guidance with

the velocity and acceleration vectors aligned [1]. Still, the acceleration, velocity, and position profiles do not satisfy the desired conditions. The most promising E Guidance extension involves higher order integration of the E Guidance functions, but it has some impractical movements initially away from the target. The initial movement away from the target may be useful for incoming obstacle avoidance only if there are no obstacles behind or below the vehicle. Numerous combinations with polynomials, exponentials, and trigonometric functions for $p_i(t), p_j(t)$ do not satisfy the boundary conditions, but some combinations partially fulfill the boundary conditions. Other combinations do not reveal real numbers, so they yield impractical solutions.

This paper describes the proposed E Guidance extensions in Ref. [2] with more details and some modifications, and it is organized as follows. Section 2 discusses the original equations, and section 3 describes methods to extend E Guidance. Section 4 provides simulation results for the proposed extensions, and finally, section 5 ends with concluding remarks.

2 Original Formulation for E Guidance

The translational commanded thrust acceleration, $a_{Tx}(t)$, in the x-direction is [1]:

$$a_{Tx}(t) = c_1 p_1(t) + c_2 p_2(t) - g_x(t) \ , \tag{1}$$

such that $p_1(t)$ and $p_2(t)$ are linearly independent functions and $c_1, c_2 \in \mathbb{R}$.

$$p_1(t) = \tau^m \ , \ p_2(t) = \tau^n \tag{2}$$

with $m, n \in \mathbb{Z}$ (integers) and $\tau = (T - t)$ such that T is the terminal time and t is the current time. The time-to-go to reach the desired position and velocity is defined by the difference between the terminal and current time: $T_{go} = T - t$. The general E Guidance procedure starts with setting the initial position and velocity boundary conditions at an initial time, t_0, and the desired position and velocity at some terminal time, T. Second, choose linearly independent functions, $p_i(t), p_j(t)$, for $i = 1, 3, 5$ and $j = 2, 4, 6$ for three dimensional E Guidance. E Guidance in one dimension requires only $p_1(t), p_2(t)$. Third, integrate $p_i(t), p_j(t)$ to determine $\mathbf{F} = [f_{ij}]$ and $\mathbf{E} = [e_{ij}]$ as the inverse of the $\mathbf{F}$ matrix:

$$\begin{aligned}
&e_{11} = f_{22}/\triangle \ , \ e_{12} = -f_{12}/\triangle, e_{21} = -f_{21}/\triangle \ , \ e_{22} = f_{11}/\triangle \ , \\
&f_{11} = \int_{t_0}^{T} p_1(t)dt \ , \ f_{12} = \int_{t_0}^{T} p_2(t)dt \ , \ f_{21} = \int_{t_0}^{T} \left[\int_{t_0}^{t} p_1(s)ds\right] dt \ , \\
&f_{22} = \int_{t_0}^{T} \left[\int_{t_0}^{t} p_2(s)ds\right] dt \ , \ \triangle = det(\mathbf{F}) = f_{11}f_{22} - f_{12}f_{21} .
\end{aligned} \tag{3}$$

Fourth, determine the coefficients, c_1, c_2 (x-direction), from the $\mathbf{E}$ matrix components and the boundary conditions:

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} \dot{x}_D - \dot{x}_0 \\ x_D - x_0 - \dot{x}_0 T_{go} \end{bmatrix} \ , \tag{4}$$

in which "0" denotes initial, and "D" denotes desired. Finally, use the computed coefficients to compute the thrust acceleration profile as shown in Eq. (1). Integrating yields the guided velocity profile, and integrating again generates the guided position profile. Figure 1 shows an example of translational E Guidance.

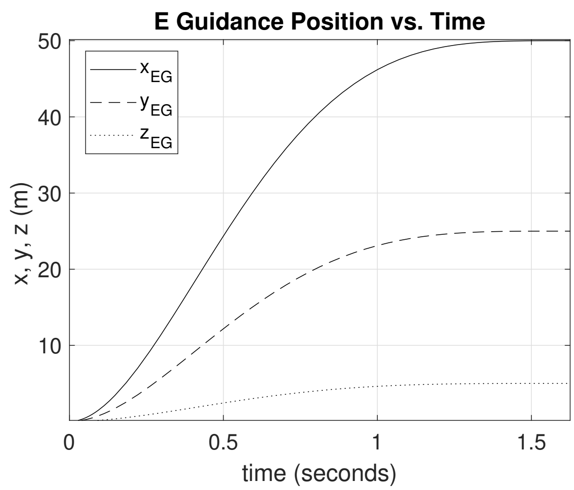


Fig. 1: Guided Trajectory Example

3 Description of E Guidance Extensions

It is possible to extend E Guidance to rotational acceleration commands [2–4]:

$$
\begin{aligned}
\alpha_\phi(t) &= c_1\tau^2 + c_2\tau^3 - \frac{I_{yy} - I_{zz}}{I_{xx}}\omega_y\omega_z \ , \\
\alpha_\theta(t) &= c_3\tau^2 + c_4\tau^3 - \frac{I_{zz} - I_{xx}}{I_{yy}}\omega_x\omega_z \ , \\
\alpha_\psi(t) &= c_5\tau^2 + c_6\tau^3 - \frac{I_{xx} - I_{yy}}{I_{zz}}\omega_x\omega_y
\end{aligned}
\tag{5}
$$

in which $\alpha_\phi, \alpha_\theta, \alpha_\psi$ are the angular acceleration commands and $\omega_x, \omega_y, \omega_z$ are the body angular velocities. See Ref. [2–4] for more details on rotational E Guidance. The next subsections discuss four translational E Guidance extensions.

3.1 Desired Intermediate Positions and Velocities

The first method considers intermediate conditions halfway during the maneuver such that $T_{go,int} = T_{go}/2$. The $\mathbf{E}$ matrix utilizes the original formulation defined by Cherry as: $\mathbf{E} = \mathbf{F}^{-1}$ [1]. Including intermediate conditions yields $\mathbf{E}, \mathbf{F} \in \mathbb{R}^{4\times4}$ with the upper

left submatrix containing the intermediate conditions and the lower right submatrix containing the endpoint (terminal) conditions:

$$\mathbf{F} = \begin{bmatrix} \mathbf{F1}_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & \mathbf{F2}_{2\times2} \end{bmatrix} \tag{6}$$

with submatrices $\mathbf{F1}, \mathbf{F2}$ defined as:

$$\begin{aligned} \mathbf{F1} &= \begin{bmatrix} \int_{t_0}^{T/2} p_1(t)dt & \int_{t_0}^{T/2} p_2(t)dt \\ \int_{t_0}^{T/2} \big(\int_{t_0}^{t} p_1(s)ds \big) dt & \int_{t_0}^{T/2} \big(\int_{t_0}^{t} p_2(s)ds \big) dt \end{bmatrix} \\ \mathbf{F2} &= \begin{bmatrix} \int_{t_0}^{T} p_1(t)dt & \int_{t_0}^{T} p_2(t)dt \\ \int_{t_0}^{T} \big(\int_{t_0}^{t} p_1(s)ds \big) dt & \int_{t_0}^{T} \big(\int_{t_0}^{t} p_2(s)ds \big) dt \end{bmatrix} \end{aligned} \tag{7}$$

The vector with the intermediate (I) and final (F) conditions for the x-direction is defined by the first two rows for the intermediate conditions and the last two rows for the final conditions:

$$\boldsymbol{x}_{I,F}(t) = \begin{bmatrix} \dot{x}(T/2) - \dot{x}(t_0) \\ x(T/2) - x(0) - \dot{x}(t_0)T_{go}/2 \\ \dot{x}(T) - \dot{x}(T/2) \\ x(T) - x(T/2) - \dot{x}(T/2)T_{go}/2 \end{bmatrix}. \tag{8}$$

with $x(0), x(T/2), x(T), \dot{x}(0), \dot{x}(T/2), \dot{x}(T)$ defined as the initial position, intermediate position, final position, initial velocity, intermediate velocity, and final velocity, respectively. Similar forms exist for the y, z directions but are not shown for brevity. Overall, the method presented here in this paper differs from the method in Ref. [2] by:

1. integrating two linearly independent polynomials instead of four linearly independent polynomials
2. using the intermediate conditions for the computations of the final conditions, i.e., last two rows of Eq. (8)

The default and simplest set of linearly independent polynomials is:

$$p_1(t) = 1\ ,\ p_2(t) = (T - t). \tag{9}$$

Testing this method with the polynomials in Eq. (9) shows that the desired final and intermediate position conditions and intermediate velocity conditions are not satisfied. However, the final velocity conditions are satisfied with zero acceleration at the end of the maneuver. Preliminary tests also demonstrate extremely large final position values such that this proposed method is not physically feasible or practical. Consequently, section 4 will not show these simulation results for brevity.

3.2 Final Desired Jerk Vector

This method augments the 2×2 $\mathbf{E}$ matrix to 4×4 with final desired acceleration and jerk values as the third and fourth set of conditions for computing the c_i. The first two conditions are the original conditions in the column vector on the right-hand-side of Eq.

(4). The jerk vector is simply the time derivative of the acceleration vector, which is straightforward to compute because acceleration is a function of only time and constants:

$$\begin{aligned} a_x &= c_1 p_1(t) + c_2 p_2(t) + c_3 p_1(t) + c_4 p_4(t) \\ j_x &= \frac{da_x}{dt} = c_1 p_1'(t) + c_2 p_2'(t) + c_3 p_1'(t) + c_4 p_4'(t) \end{aligned}. \tag{10}$$

This method algebraically solves for the coefficients without utilizing the **E** or **F** matrices, i.e., four coefficients (unknowns) and four equations: two from the column vector on the right-hand-side of Eq. (4) and two from Eqn. (10). This approach follows Cherry's method for final desired attitude guidance by aligning the final velocity and acceleration vectors [1]. Overall, preliminary tests for this method yield drastic divergence and does not satisfy the boundary conditions. Therefore, section 4 will not show these simulation results.

3.3 Four Integrations of the F Matrix

This method differs from the previous method by including the **E** and **F** matrices by utilizing four integrations for four polynomials:

$$\begin{aligned} \mathbf{F} &= \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \\ f_{41} & f_{42} & f_{43} & f_{44} \end{bmatrix} \\ &= \begin{bmatrix} \int p_1(t)dt & \dots & \int p_4(t)dt \\ \int\int p_1(s)dsdt & \dots & \int\int p_4(s)dsdt \\ \int\int\int p_1(u)dudsdt & \dots & \int\int\int p_4(u)dudsdt \\ \int\int\int\int p_1(r)drdudsdt & \dots & \int\int\int\int p_4(r)drdudsdt \end{bmatrix}. \end{aligned} \tag{11}$$

The third condition of **E** is final desired acceleration, and the fourth condition is the final desired jerk. Here is the equation for computing the coefficients for the x-direction:

$$\mathbf{c}_x = \begin{bmatrix} e_{11} & e_{12} & e_{13} & e_{14} \\ e_{21} & e_{22} & e_{23} & e_{24} \\ e_{31} & e_{32} & e_{33} & e_{34} \\ e_{41} & e_{42} & e_{43} & e_{44} \end{bmatrix} \begin{bmatrix} \dot{x}_D - \dot{x}_0 \\ x_D - x_0 - \dot{x}_0 T_{go} \\ a_{x_D} + g_x \\ j_{x_D} \end{bmatrix}. \tag{12}$$

The final desired position, velocity, and jerk conditions are not satisfied, but the final desired acceleration is. Thus, section 4 does not show these simulation results.

3.4 Three Integrations of the F Matrix

This method resembles the previous method but integrates three polynomials such that the rows of the **F** matrix have one, two, or three integrations for $p_1(t), p_2(t), p_3(t)$:

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} = \begin{bmatrix} \int p_1(t)dt & \int p_2(t)dt & \int p_3(t)dt \\ \int\int p_1(s)dsdt & \int\int p_2(s)dsdt & \int\int p_3(s)dsdt \\ \int\int\int p_1(u)dudsdt & \int\int\int p_2(u)dudsdt & \int\int\int p_3(u)dudsdt \end{bmatrix} \tag{13}$$

The third condition involves the final desired acceleration such that the coefficients for the x-direction are:

$$\mathbf{c}_x = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} \dot{x}_D - \dot{x}_0 \\ x_D - x_0 - \dot{x}_0 T_{go} \\ a_{x_D} + g_x \end{bmatrix} . \tag{14}$$

The other coefficient vectors, $\mathbf{c}_y$ and $\mathbf{c}_z$, have similar forms and are not shown for brevity. This method demonstrates promising results since it satisfies the position and velocity boundary conditions. However, the final desired acceleration is not satisfied. Overall, since this method has the best results of the four methods, section 4 shows the simulation results for this method.

4 Simulation Results

This section shows simulation results for the proposed extension with three polynomials for three UAV maneuvers: a $360°$ roll maneuver, a vertical one-dimensional takeoff maneuver, and a waypoint maneuver.

4.1 Roll Maneuver

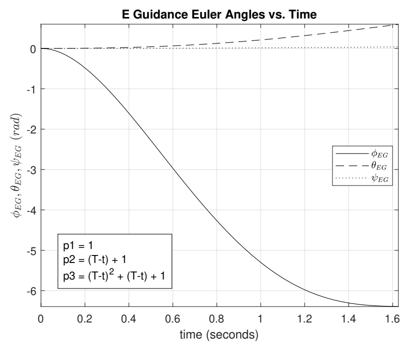
The coefficients, c_1, c_2, c_3, resemble the original 2×2 expression for the $\mathbf{E}$ matrix but with desired angular acceleration as the third element in the vector multiplied by the $\mathbf{E}$ matrix:

$$\begin{aligned} c_\phi &= \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} \dot{\phi}_D - \dot{\phi}_0 \\ \phi_D - \phi_0 - \dot{\phi} T_{go} \\ \ddot{\phi}_D \end{bmatrix} , \\ c_\theta &= \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} \dot{\theta}_D - \dot{\theta}_0 \\ \theta_D - \theta_0 - \dot{\theta} T_{go} \\ \ddot{\theta}_D \end{bmatrix} , \\ c_\psi &= \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} \dot{\psi}_D - \dot{\psi}_0 \\ \psi - \psi - \dot{\psi} T_{go} \\ \ddot{\psi}_D \end{bmatrix} . \end{aligned} \tag{15}$$

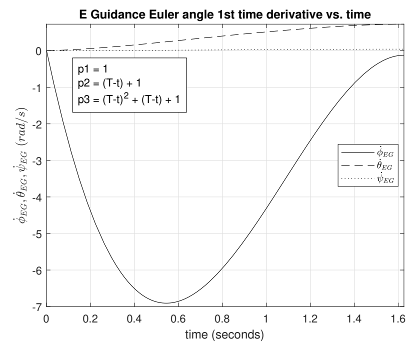
For three polynomials, the boundary conditions for a $360°$ roll maneuver are:

$$\begin{aligned} \phi_D = -2\pi \ , \ \dot{\phi}_D = 0 \ , \ \theta_D = 0 \ , \\ \dot{\theta}_D = 0 \ , \ \psi_D = 0 \ , \dot{\psi}_D = 0 \ , \\ \ddot{\phi} = 0 \ , \ \ddot{\theta} = 0 \ , \ \ddot{\psi} = 0. \end{aligned} \tag{16}$$

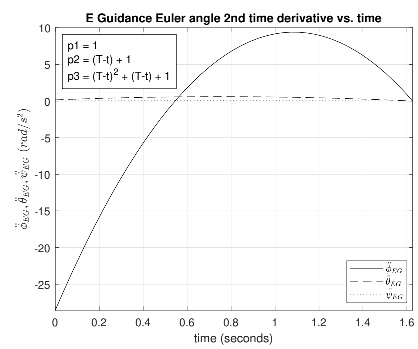
Figure 2 shows the Euler angles and time derivatives for a $360°$ roll maneuver for a quadcopter UAV with polynomials: $p_1(t) = 1, p_2(t) = (T - t) + 1, p_3(t) = (T - t)^2 + (T - t) + 1$. The pitch angle diverges and has a final pitch angle of 0.595 rad or $34.1°$, which is not even close to the final desired pitch angle of $0°$.



(a) Guided 360° Roll Maneuver: Euler Angles



(b) Guided 360° Roll Maneuver: 1st Time Derivative



(c) Guided 360° Roll Maneuver: 2nd Time Derivative

Fig. 2: Guided Roll Maneuver: $p_1(t) = 1, p_2(t) = (T-t)+1, p_3(t) = (T-t)^2 + (T-t) + 1$

4.2 Takeoff Maneuver

This subsection describes a one-dimensional vertical ascent (takeoff maneuver) to stop at a desired altitude of 15 m. The polynomial selection is:

$$p_1(t) = 1\ ,\ p_2(t) = (T-t) + 1\ ,\ p_3(t) = (T-t)^2 + (T-t) + 1. \tag{17}$$

Figure 3 shows the guided takeoff maneuver with the altitude, vertical velocity, and vertical acceleration. The guided takeoff is very smooth, and the largest speed is approximately $0.85\ m/s$. Decreasing the time-to-go will increase the magnitude of the velocity profile because the vehicle has less time to reach the desired altitude. The acceleration is nearly constant at around $10\ m/s^2$ due to gravity. The final altitude is $15\ m$ from the initial height, and the final vertical velocity is $0\ m/s$, which satisfies the final desired boundary conditions.

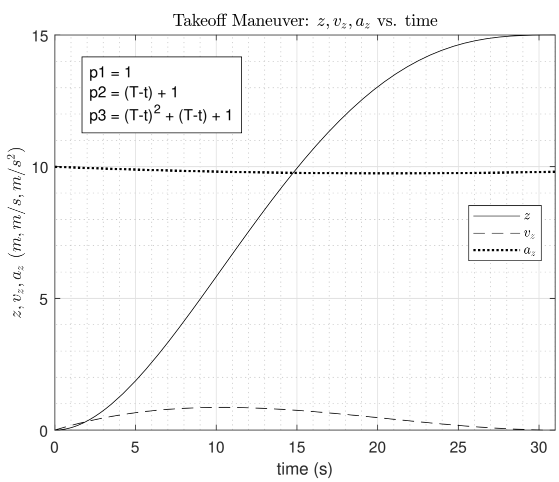
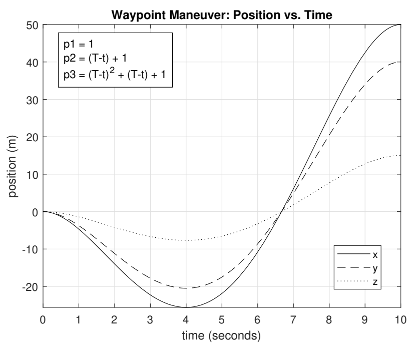


Fig. 3: Guided Takeoff Maneuver

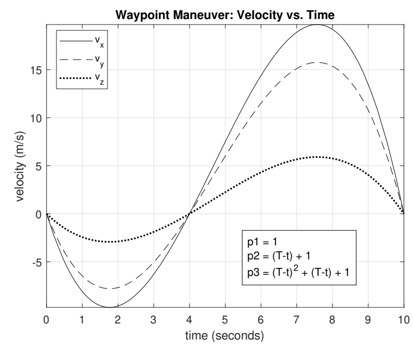
4.3 Waypoint Maneuver

This subsection describes a three-dimensional waypoint maneuver for a vehicle to stop at a waypoint located 50 m away in the x-direction, 40 m away in the y-direction, and increases its altitude (z-component) by 15 m. The choice for $p_1(t), p_2(t), p_3(t)$ is the same as the polynomial selection in Eq. (17). Figure 4 shows the guided waypoint trajectory profile. Figure 4a shows the trajectory, which ends at the intended waypoint. It has an initial backward motion away from the target for approximately the first 4 seconds before moving toward the target. This initial reverse may not be practical or physically possible if there are obstacles behind the vehicle. Figure 4b shows the velocity profile, which shows a max reverse speed of around -10 m/s at approximately

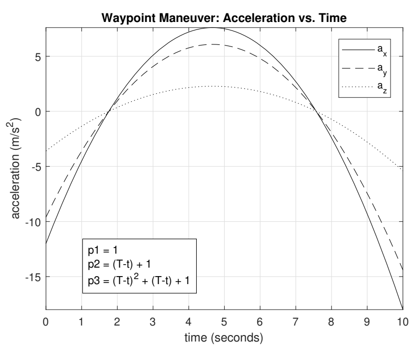
1.7 seconds, a max forward speed of around 20 m/s at approximately 7.5 seconds, and satisfaction of the initial and final velocity conditions, i.e., 0 m/s. Figure 4c shows the acceleration profile, and the final accelerations are nonzero. Appendix A of Ref. [2] has



(a) Guided Waypoint: Position vs. Time



(b) Guided Waypoint: Velocity vs. Time



(c) Guided Waypoint: Acceleration vs. Time

Fig. 4: Guided Waypoint Maneuver: $p_1(t) = 1, p_2(t) = (T-t)+1, p_3(t) = (T-t)^2 + (T-t)+1$

numerous possible combinations of exponential, sinusoidal, and polynomial functions for $p_1(t), p_2(t0 < p_3(t)$. The trigonometric and exponential functions perform poorly by diverging, oscillating, or not satisfying the boundary conditions.

5 Conclusion

In summary, there were four methods for extending E Guidance. The first method utilized intermediate and final conditions with a sparse $\mathbf{E}$ matrix and four functions: $p_1(t), p_2(t), p_3(t), p_4(t)$. The second method utilized final desired jerk and followed Cherry's formulation for final desired attitude without including $\mathbf{E}$. The third method resembled the second method but included four integrations in $\mathbf{F}$. The fourth method

used three integrations for the three functions, $p_1(t), p_2(t), p_3(t)$, and included final desired acceleration.

The fourth method is the most promising since it satisfies the position and velocity boundary conditions but not the final desired acceleration. However, the fourth proposed method has an initial reverse, which might not be practical due to vehicle motion constraints, actuator dynamics, or obstacles behind the vehicle. For instance, fixed-wing aircraft cannot reverse in mid-flight, but multicopter UAVs can reverse in mid-flight. Conversely, Cherry's original formulation with just two functions lacks the initial backward motion and is more practical. A potential application for this extended E Guidance method involves avoiding incoming obstacles while the initial plan was to move forward. In this case, applying E Guidance discretely with regular navigation and obstacle detection updates ensures the dynamic obstacle is out of the vehicle's intended path before proceeding forward.

Overall, extending E Guidance to three or four functions does not yield better performance than Cherry's original formulation with $p_1(t), p_2(t)$. Increasing the exponents to higher powers typically generates higher accuracy like a Taylor series expansion, but that is not the case for the proposed E Guidance methods. It is possible that there are other combinations for the functions that satisfy all the boundary conditions, so the authors might not have been lucky in their selection of the functions. Even though three of the four methods do not satisfy the boundary conditions, they are presented here to prevent future researchers from reinventing inadequate methods, which will save time researching innovative methods for extending E Guidance.

References

1. Cherry, G.: A general, explicit, optimizing guidance law for rocket-propelled spaceflight. In: Astrodynamics Guidance and Control Conference, p. 638 (1964)
2. Kawamura, E.: Integrated targeting, guidance, navigation, and control for unmanned aerial vehicles. Ph.D. dissertation, University of Hawai'i at Manoa (2020)
3. Kawamura, E., Azimov, D.: Integrated targeting, guidance, navigation, and control for unmanned aerial vehicles. In: Volume 168 of the Advances in the Astronautical Sciences Series, pp. 4259–4277. Univelt (2019)
4. Kawamura, E., Azimov, D.: Extremal control and modified explicit guidance for autonomous unmanned aerial vehicles. Journal of Autonomous Vehicles and Systems **2**(1) (2022)